Engineering Optimization

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/geno20

Optimizing the maintenance schedule for a vehicle fleet: a simulation-based case study

Yali Wang, Steffen Limmer, Duc Van Nguyen, Markus Olhofer, Thomas Bäck & Michael Emmerich

To cite this article: Yali Wang, Steffen Limmer, Duc Van Nguyen, Markus Olhofer, Thomas Bäck & Michael Emmerich (2021): Optimizing the maintenance schedule for a vehicle fleet: a simulation-based case study, Engineering Optimization, DOI: 10.1080/0305215X.2021.1919888

To link to this article: https://doi.org/10.1080/0305215X.2021.1919888

ENGINEERING OPTIMIZATION
https://doi.org/10.1080/0305215X.2021.1919888

Taylor & Francis
Taylor & Francis Group

Optimizing the maintenance schedule for a vehicle fleet: a simulation-based case study

Yali Wang[a], Steffen Limmer[b], Duc Van Nguyen[a], Markus Olhofer [b], Thomas Bäck[a] and Michael Emmerich[a]

[a]Leiden Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands; [b]Honda Research Institute Europe GmbH, Offenbach, Germany

ABSTRACT

Vehicle fleets support a diverse array of functions and are increasing rapidly in the world of today. For a vehicle fleet, maintenance plays a critical role. In this article, an evolutionary algorithm is proposed to optimize the vehicle fleet maintenance schedule based on the predicted remaining useful lifetime (RUL) of vehicle components to reduce the costs of repairs, decrease maintenance downtime and make them safer for drivers. The multi-objective evolutionary algorithm (MOEA) is then enhanced to focus precisely on the preferred solutions. Moreover, stability is involved as another objective in the dynamic MOEA for handling the problem under changes in the environment. To implement the complete maintenance process, a simulator is developed that can define vehicles, predict the RUL of components and optimize the maintenance schedule in a rolling-horizon fashion. The results of the proposed MOEAs under different scenarios are reported and compared.

ARTICLE HISTORY
Received 4 February 2021
Accepted 10 April 2021

KEYWORDS
Multi-objective evolutionary optimization; preference-based optimization; dynamic optimization; prognostic-based maintenance scheduling

1. Introduction

Maintenance scheduling is the management and allocation of maintenance tasks. It should be decided when the maintenance tasks are conducted and by whom. Optimization plays a critical role in the efficient usage of the equipment or machinery being maintained. The optimization of maintenance schedules can yield significant savings for companies. Additionally, it can ensure worker safety, extend system and component life and significantly reduce the probability of catastrophic failure. This article addresses the problem of optimizing the maintenance schedules for a vehicle fleet. A vehicle fleet is usually a commercial fleet used to transport people or goods, such as a trucking fleet, delivery fleet, taxi fleet, car rental fleet, public utility fleet, and so on. For a vehicle fleet, a good maintenance schedule can reduce related expenses, increase resource utilization, ensure consistent service delivery, and even reduce its carbon footprint. Given the fact that multiple objectives need to be achieved simultaneously, a multi-objective vehicle fleet maintenance scheduling optimization (MOVFMSO) problem is considered in this article.

Recently, the remaining useful lifetime (RUL), *i.e.* the duration of a component to no longer meet operational requirements, has been used in maintenance optimization due to the development of predictive models, *e.g.* in maintenance and mission planning (Camci *et al.* 2019). For the optimization of the vehicle fleet maintenance schedule, the RUL can also be used since damage to vehicles

CONTACT Yali Wang y.wang@liacs.leidenuniv.nl .

Supplemental data for this article can be accessed at https://doi.org/10.1080/0305215X.2021.1919888

occurs after varying durations. To be more precise, each vehicle consists of several components that have to be maintained based on their respective damage from wear and tear. To identify when each component should be maintained, the RUL of each component can be predicted based on predictive approaches or models (Xia *et al.* 2018; Van Nguyen *et al.* 2019). Additionally, to maintain components, different maintenance resources are provided, for example, repair workshops. The solution to solving the application problem is to assign maintenance activities to workshops based on the predicted RUL and the known maintenance resources.

Metaheuristics are suggested here to solve this problem, since the combinatorial and nonlinear nature of the problem does not allow for an exact solution. Among all metaheuristics, evolutionary algorithms (EAs) have been the most popular in the last few decades, giving rise to a category known as multi-objective evolutionary algorithms (MOEAs) (Thu Bui and Alam 2008). In this article, a tailored MOEA is applied to find the Pareto optimal set of the MOVFMSO problem. Since all solutions on the Pareto front (PF) are considered equally good, but only one solution (schedule) can be deployed in practice, the proposed MOEA is then extended to a preference-based MOEA in order to find preferred solutions, accelerate convergence and pick the final optimal solution.

To model the complete process of maintaining the vehicle fleet by way of scheduling optimization, a simulator is developed that starts from simulating driving tasks, a vehicle fleet and available workshops. The RULs of components are predicted when the vehicles execute the distributed driving tasks. Based on the RUL, the maintenance schedule is optimized by the MOEA for the fleet. Owing to the requirement for updating the maintenance schedule periodically, the problem is further upgraded to a dynamic optimization problem. In the simulator, the optimization process runs in a rolling-horizon fashion and a new maintenance schedule is generated periodically based on the newly predicted RULs. Accordingly, the optimization algorithm becomes a dynamic algorithm and a fourth objective is added into the dynamic MOEA, which is to minimize the changes between the new schedule and the previous schedule.

This article is organized as follows: Section 2 presents the details of the proposed optimization objectives and model. Section 3 introduces the multi-objective optimization methods. Section 4 discusses the details of the simulator and Section 5 shows the experimental results. Section 6 concludes the article.

2. Problem definition and modelling

2.1. Optimization problem

For a vehicle fleet running driving tasks, vehicle components are becoming damaged and should be maintained regularly. Several separate workshops are available for maintenance of the vehicle fleet, and the repair time and maintenance cost are known for each component in each workshop. In addition to the time and cost for repairing each vehicle component, a fixed set-up cost and set-up time are considered for each visit of a vehicle to a workshop, which correspond to the cost and time required for the preparation of the maintenance operation. Three objectives are assumed to be relevant for the vehicle fleet operator, which are the minimization of the total workload, the total maintenance cost and the expected number of failures when vehicles break down on the road. Clearly, these objectives conflict with each other. For instance, more frequent maintenance leads to a reduction of failures, but increases maintenance cost and workload.

In a real-world scenario, after a maintenance schedule is released for execution, continuous updating of the schedule is required owing to changes in the condition of the vehicle and the ensuing changes in the RUL predictions. The optimization of the maintenance schedule is an ongoing process and it is therefore desirable to generate robust schedules. When the proposed static algorithm is extended to a dynamic algorithm, a fourth objective is involved in the algorithm, which is the stability, *i.e.* the start and the completion of each activity should be as close as possible to its previous schedule.

The mathematical formulation of the problem can be found in the online supplemental data for this article.

2.2. Remaining useful lifetime prediction

Knowing the RUL is essential for establishing an optimal maintenance schedule, and the RUL prediction provides the system residual life from its current condition and the past operation profile (Vachtsevanos *et al.* 2006). Commonly, approaches used in prognostics and predicting RUL are classified into three types: physics-based approaches, data-driven approaches and hybrid approaches (Elattar, Elminir, and Riad 2016). In this article, physics-based models are used to estimate the degradation and failures of four essential components of a vehicle, namely the engine, brake pads, helical springs and tyres. The fatigue and wear mechanisms are well established for these components.

2.2.1. Degradation of helical springs

The helical spring is the most common type of spring used in passenger vehicles. The main mechanism that reduces the lifetime of a helical spring is fatigue and it is often analysed using the S–N curve, which describes the relation between cyclic stress amplitude and number of cycles to failure (Tinga 2013). Figure 1 shows a typical S–N curve. The vertical axis shows the stress amplitude, whereas the horizontal axis indicates the corresponding number of cycles to failure at a given stress amplitude. A stress S is calculated from a force F by the equation

$$S = K\frac{8 \times F \times D_{\text{coil}}}{\pi \times d_{\text{wire}}^3},$$

where D_{coil} and d_{wire} are the mean diameters of the coil and the wire, respectively. $C = D_{\text{coil}}/d_{\text{wire}}$ is the spring index. $K = 1 + 0.5/C$ is the so-called Wahl factor. According to the Paris–Erdogan and Palmgren–Miner laws (Sobczyk and Spencer 1993), the percentage damaged of a spring can be formulated as

$$d_s = \sum_{i=1}^{p} \frac{n_i}{N_i} \times 100\%,$$

where d_s is the total percentage of life consumed, p is the total number of the considered stress sources, n_i and N_i are the number of cycles with a stress amplitude and the corresponding number of cycles to failure at this stress with $i = 1, 2, ..., p$ from p sources. $\frac{n_i}{N_i}$ is the fractional damage received from the ith source. When $d_s \geq 100\%$, the spring's lifetime ends and a spring failure occurs.

2.2.2. Degradation of brake pads

A worn-out failure arises as a result of cumulative damage related to loads applied over an extended time. In the process of braking, due to friction between the surfaces of the friction couple, the zones of contact are damaged after each braking event, resulting in worn-out material. The volume of the worn-out material of the ith braking event can be represented as $\Delta V_{bi} = C_{\text{brake}} \times F_i \times \Delta d_i$, where C_{brake} is a constant and represents the brake pad quality, and F_i and Δd_i are the friction force and the relative displacement between the brake pad and the brake rotor of the ith braking event, respectively. If V_{b0} is the maximum volume by which the brake pad can be reduced before a failure might occur, the percentage of the brake pad damaged (d_b) can be estimated by

$$d_b = \sum_{i=1}^{n} \frac{\Delta V_{bi}}{V_{b0}} \times 100\%.$$

The brake force is converted from the brake torque by dividing the torque by the length of the level arm. For the values of parameters in the physical models, such as D_{coil}, d_{wire}, C_{brake}, C_{tyre}, C_{engine}, *etc.*, the reader is refered to Van Nguyen *et al.* (2019).

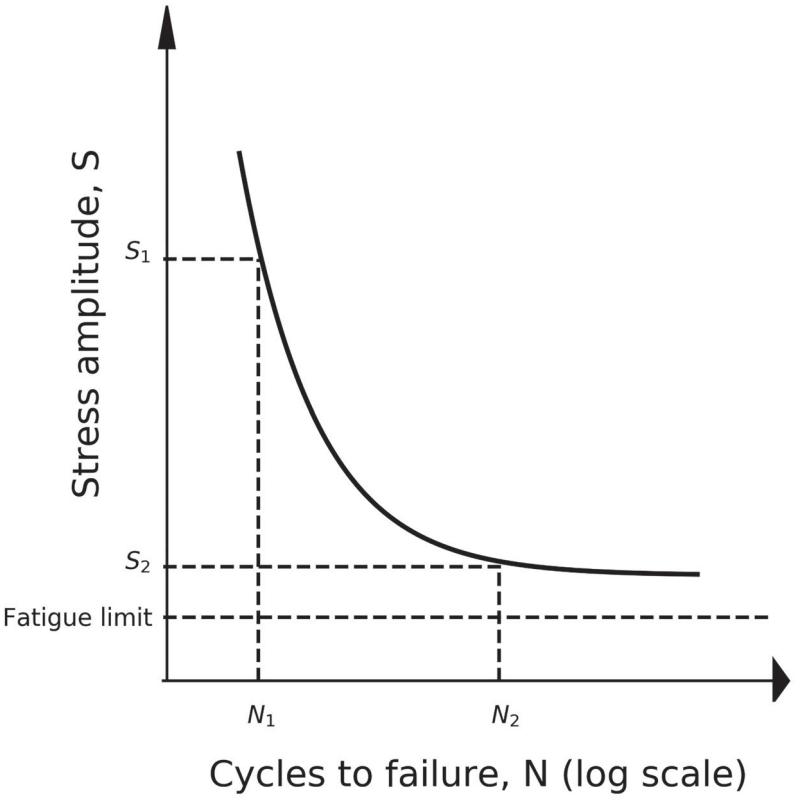

Figure 1. A typical S–N curve.

2.2.3. Degradation of tyres

The wear mechanism also applies to tyres because tyre surfaces are in contact with the road surface and friction results in worn-out tyre material. The two horizontal components of the force that cause tyres to wear out are F_x and F_y. The vertical force component F_z is only considered for pressure (overinflation, underinflation) damage to the tyres. Similarly, the volume reduction of a tyre due to worn-out material is formulated as

$$\Delta V_{ti} = C_{\text{tyre}} \times (|F_x| + |F_y|) \times \Delta d_i,$$

where C_{tyre} is a constant and represents the tyre quality. Δd_i is the relative displacement between the tyre surface and the road surface and is simply the vehicle's distance travelled. Again, the percentage of the tyre damaged (d_t) can be computed by

$$d_t = \sum_{i=1}^{n} \frac{\Delta V_{ti}}{V_{t0}} \times 100\%,$$

where V_{t0} is the maximum volume by which the tyre can be reduced before a failure might occur.

2.2.4. Degradation of vehicle engine

A rough model is established to estimate the consumption lifetime of the vehicle engine from the distance travelled and the engine rotation speed. The equation is $d_{ei} = C_{\text{engine}} \times \Delta d_i \times R_i$, where C_{engine} is a constant and represents the engine quality. Here Δd_i and R_i are the vehicle travel interval and the engine rotation speed corresponding to this travel interval, respectively. The consumed

lifetime percentage of the engine is $d_e = \sum_{i=1}^{n} d_{ei} \times 100\%$. The engine needs to be maintained if the d_{ei} sum up to one.

2.2.5. RUL calculation

It is assumed that the physical models are accurate; therefore, the real damage to components up to now can be diagnosed. The RUL is predicted by extrapolating the future damage from the distribution of the damage so far. The RUL of a component can be calculated based on the percentage damaged. If the RUL is estimated in units of one week, the total percentage damaged after the wth week is calculated by

$$D = \sum_{i=1}^{w} D_i,$$

where D_i is the total percentage damaged by the end of the ith week. Thus, the RUL after week w can be estimated by

$$\text{RUL} = \frac{100\% - D}{D/w}.$$

Here, 100% means that the component is absolutely new in the beginning. A Gaussian distribution is fitted to the distribution of the weekly percentage damaged and the resulting standard deviation σ is used to calculate the lower and upper bounds of the standard deviation confidence interval of RUL as follows:

$$\text{RUL}_{-} = \frac{100\% - D}{D/w + \sigma} \quad \text{and} \quad \text{RUL}_{+} = \frac{100\% - D}{D/w - \sigma}.$$

3. Optimization methods

Based on the predicted RUL probability distribution, *i.e.* a Gaussian distribution with mean μ and standard deviation σ, for each component its execution window is defined as $[\mu - 2 \times \sigma, \mu + 2 \times \sigma]$. The maintenance of a component can only start from a time point in its execution window. At the same time, if the maintenance times of multiple components of the same vehicle are close, *i.e.* if their execution windows overlap, these components can be grouped together as one maintenance activity to save commuting time and facilitate the preparation of workshops, *i.e.* reduce the set-up time and set-up cost in the algorithm. When developing MOEA to solve the MOVFMSO problem, a three-vector chromosome is designed to define a solution. The component grouping, starting times and executing workshops of the maintenance activities are represented by three vectors, respectively.

The component grouping vector tells which components are to be combined into the same maintenance activity. It is initialized by randomly picking a feasible group structure for each vehicle. According to the component combination, the starting time and workshop of a maintenance activity can be initialized by randomly picking a time spot in its overlapping execution window and a workshop. In accordance with the problem and its encoding, specific crossover and mutation operators have been designed for the problem. The reader is referred to Wang, Limmer, *et al.* (2019) for the details.

In EAs, each chromosome in a generation needs to be evaluated for the selection process and this is accomplished by converting a chromosome into a feasible schedule to calculate the objective values. The first objective, *i.e.* the total workload, consists of two parts: the set-up time and the maintenance time. For one maintenance activity, the set-up time of the vehicle is counted only once and the maintenance time is the sum of the processing times of all components within the activity. The total cost objective consists of three parts: the set-up cost, the maintenance cost and the penalty cost. Again, for one maintenance activity, the set-up cost of the vehicle is counted only once and the maintenance cost is the sum of the repairing costs of all components within the activity. The penalty cost is used

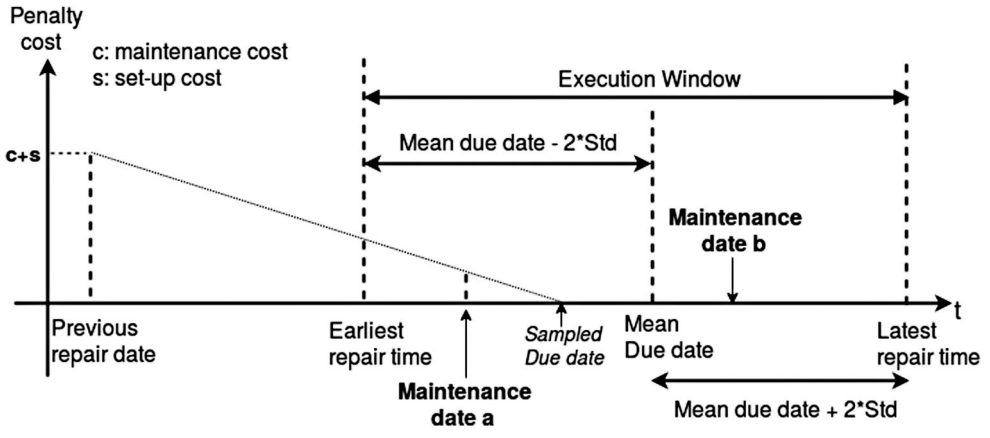

Figure 2. Execution window of a component.

to reflect the waste of useful time when the component is maintained earlier than the date when the component would break down without maintenance, which is also called the 'due date'.

Monte Carlo simulation is adopted to account for uncertainties in the due date; the penalty cost and the third objective (the expected number of failures) are also calculated in the process. For each component, 1000 samples of the due date are generated in its execution window, based on a Gaussian distribution. The scheduled maintenance date of the component is compared with these samples one by one. When the scheduled maintenance date is earlier than a sample, it can be assumed that the component will be maintained before it has broken down. In this case, the useful life between the maintenance date until it would have broken down will be wasted and a corresponding penalty cost is computed for the waste. To calculate the penalty cost, a linear penalty function is suggested based on the following assumptions.

- If a component is maintained when it is new or the previous maintenance has just been completed, the penalty cost is the full cost of maintaining it: the maintenance cost of the component plus the set-up cost for the vehicle;
- If a component is maintained exactly on its due date, the penalty cost is zero.

Figure 2 shows the execution window of a component. Given a 'sampled due date', the oblique dotted line is generated as the linear penalty function. Given this, the penalty cost or the expected number of failures of this 'sampled due date' is calculated depending on whether its scheduled maintenance date is earlier or later than that of the sample. When the scheduled maintenance date is located on the left side of the 'sampled due date', *e.g.* 'Maintenance date a', the corresponding penalty cost can be calculated according to the penalty function. On the other hand, when the scheduled maintenance date is later than that of the sample, *e.g.* 'Maintenance date b', it is assumed that the maintenance date is too late and a defect occurs while in use. In this case, the number of failures increases by one. The average of the penalty costs and number of failures over 1000 samples are used as the penalty cost and expected number of failures for the component.

The diversity-indicator based multi-objective evolutionary algorithm (DI-MOEA) (Wang, Emmerich, *et al.* 2019) is applied in order to solve the MOVFMSO problem using the described representation. As an indicator-based MOEA, in addition to non-dominated sorting as the first ranking criterion, DI-MOEA uses the Euclidean-distance based geometric mean gap indicator as the quality indicator to guide the search towards a uniformly distributed PF approximation regardless of the shape of the PF.

3.1. Preference-based optimization

Owing to the NP-hard nature of the MOVFMSO problem, it is not possible to find its true PF in a reasonable time. The objective space of the problem spreads over a large area, but not all the optimal solutions on the PF are preferred solutions. Therefore, the basic optimization algorithm is extended to a preference-based MOEA to focus only on preferred solutions. Usually, an interactive algorithm is used to incorporate preference information into evolutionary multi-objective optimization and it is implemented by asking the decision makers (DMs) for preference information at each iteration (Thiele *et al.* 2009). However, inspecting and choosing solutions from the whole PF is not a trivial task for the DMs. The visualization of high-dimensional space further aggravates the difficulty. An automatic preference-based MOEA is proposed to avoid these difficulties; it can generate the preference region automatically, narrow down the feasible objective space progressively, and eventually obtain the preferred solutions.

The automatic preference-based algorithm is developed by adding an extra ranking criterion in the DI-MOEA. To be specific, in the preference-based DI-MOEA, non-dominated sorting is used as the first ranking criterion; the diversity indicator, *i.e.* the Euclidean-distance based geometric mean gap indicator, is used as the second ranking criterion to ensure the uniformity of the solution set; the third ranking criterion used is Euclidean distance to the knee point (Das 1999) and the third ranking criterion is used only after the preference region has been generated. The algorithm is divided into two stages by dividing the computing budget into two parts. The first stage is a teach-in phase, in which the first half budget is used to obtain a rough entire PF. The second stage adapts to generate and narrow down the preference region step by step. The reader is referred to Wang *et al.* (2021) for more details of the preference-based algorithm.

3.2. Dynamic optimization

After achieving a PF approximation, the knee point is picked as the final optimal schedule to be deployed in workshops. In the real-world application, the maintenance schedule needs to be updated periodically. To generate a new schedule for the next stage, the current schedule is also needed. Various disruptions may occur while running a maintenance schedule; for example, the vehicle might break before its scheduled maintenance time, or new repairing tasks in workshops might lead to a delay in the scheduled activities. In the face of various disruptions, adjustments to the schedule have to be made and this is also the reason that the maintenance schedule is updated periodically. A new schedule with the new arrangement of the maintenance activities is generated from the new condition of the vehicle fleet and workshops. However, changes to the current schedule lead to additional costs such as the cost of reallocating tools and equipment, the cost of reordering raw materials, *etc.* To reduce these costs when updating the maintenance schedule, one important point is to maximize the similarity between the new schedule and the previous one to increase stability. For this purpose, a stability criterion is employed as one more objective in the dynamic algorithm.

The number of components assigned to different maintenance times or workshops from those in the current running schedule is minimized by the dynamic algorithm. Furthermore, since the stability of maintenance activities in the near future is more important than that of maintenance activities in the distant future, when calculating the stability objective, different weights are given to the components that are scheduled to be maintained within one week, within one month and beyond one month. The dynamic algorithm makes it possible for the maintenance schedule to be optimized under different operational environments including dynamic and changing conditions. Most importantly, the dynamic algorithm updates the maintenance schedule based on the latest damage to components because the underlying predicted RUL of each component is based on the latest damage. In this way, the maintenance schedule becomes more accurate.

4. Simulator

A simulator has been developed to implement and evaluate the complete process of vehicle fleet maintenance scheduling optimization. In the simulator, CarMaker[1] is adopted to simulate the driving scenarios of a taxi fleet in New York City. The origin and destination coordinates from the New York Green Taxi Corp. in January 2015 downloaded from NYC Open Data[2] are converted into taxi routes using the Google API. Four thousand trips have been simulated as the driving tasks. An extra load of between zero and 100 kg was randomly chosen and added to each passenger seat. Each vehicle was assigned to 40 random trips per day on average, and the maximum number of trips each vehicle could execute per day was 50. These trips were randomly selected from 4000 simulated trips. The sensor data on forces, braking torque and engine rotation speed yielded by CarMaker were used to estimate the percentage damaged and the RUL of springs, tyres, brake pads and engine by the physical models as described in Section 2.2.

Some parameters could be pre-defined to determine the MOVFMSO problem before running the simulator: the scale of the problem, *e.g.* the number of vehicles, components, workshops; the maintenance costs and times in workshops; the duration of running the simulator; and the frequency of updating the maintenance schedule. After running the simulator for the defined period, the following results were reported by the simulator.

- The number of defects: when a defect occurs, *i.e.* a component breaks down before the scheduled maintenance date, the number of defects increases by one.
- The total cost: besides the set-up cost and the maintenance cost, the waste of component lifetime has also been transferred to a cost and included in the total cost; this cost is called the 'too-early maintenance cost'. Unlike the penalty cost in the optimization algorithm, the 'too-early maintenance cost' in the simulator is an actual value because it is assumed that the physical models are 100% accurate and the due dates of the components calculated by them are used as the unalloyed truth. When a component is maintained based on the maintenance schedule, the simulator can calculate its current percentage damaged by the corresponding physical model and the remaining *health percentage* is transferred to a cost to reflect the waste of useful lifetime. The 'too-early maintenance cost' is calculated by the formula: remaining *health percentage* $\times$ maintenance cost of the component.
- The total maintenance time: the simulator records all the days that the vehicles cannot work, the reason being either a scheduled maintenance activity or a defect.
- The number of changed schedules: every time a maintenance schedule is updated, the number of components that have a different maintenance date or workshop is recorded.
- The number of unsatisfied trips: when a vehicle cannot execute its tasks, *e.g.* it is being maintained in a workshop, its assigned tasks will be distributed to other available vehicles, but the maximum number of tasks a vehicle can execute each day is 50. The tasks that cannot be satisfied are counted as unsatisfied trips.
- The number of scheduled maintenance activities: when a maintenance activity is executed based on the maintenance schedule, the number of scheduled maintenance activities increases by one.

Figure 3 shows the workflow of the simulator, which is executed on a daily basis. At the beginning of each day, vehicles in workshops are checked and sent back to work if the maintenance has been completed, meaning the damage to the components is set to zero. Next, the damage to each component is investigated and vehicles are sent to workshops when defects occur (meaning the percentage damaged of a component reaches 100%). In the case of a defect, the vehicle is sent to a random workshop. Afterwards, the maintenance schedule is checked and vehicles are sent to their assigned workshops if they are assigned to be maintained on that day. Thereafter, the driving trips of that day are assigned to the available vehicles and the damage to components updated. Lastly, when it is the day to generate a new maintenance schedule, the RUL distributions of components are predicted,

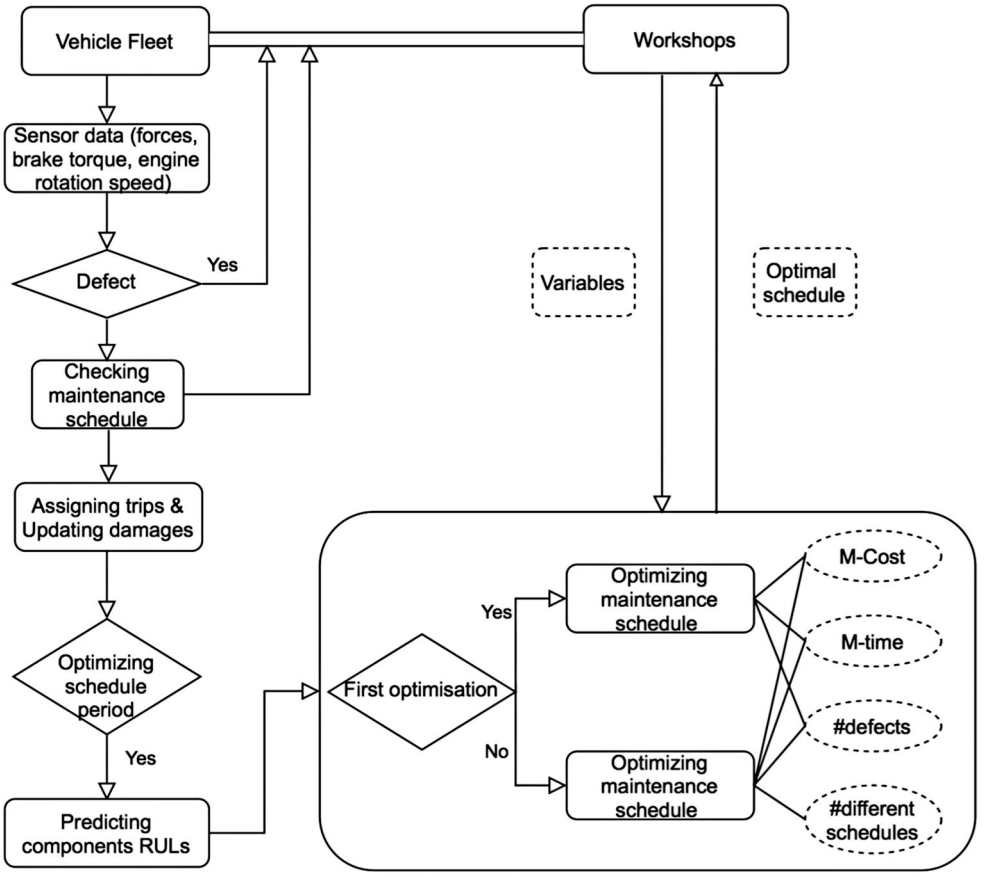

Figure 3. Daily workflow of the simulator.

and the maintenance schedule is optimized. In the case of generating the first maintenance schedule, only three objectives are employed. Later, the stability of the schedule is involved in the optimization procedure as an extra objective. After obtaining the PF approximation from each optimization, the knee point on the PF is picked and is deployed as the new schedule to replace the current schedule for maintaining the vehicle fleet.

5. Experiments

5.1. Experimental design

To show and observe the impact of different maintenance strategies clearly, the simulator runs under scenarios with the following combinations of parameters:

- the simulation time: 700 days;
- the size of the vehicle fleet: 20 vehicles with 2 workshops, 13 components for each vehicle; 20 vehicles with 5 workshops, 13 components for each vehicle;
- the frequency of schedule updating—weekly or monthly;
- the optimization computing budget—100,000 or 500,000;
- the optimization algorithm: basic MOEA; preference-based MOEA; dynamic basic MOEA; dynamic preference-based MOEA.

The results of the prediction-based optimization algorithms have also been compared with fixed-interval maintenance scheduling. To set the fixed-interval maintenance, firstly the simulator is run without the maintenance schedule. In this case, each component breaks when its damage reaches 100%, then it is maintained and sent back to work again. The average mileages are obtained for 13 components to be maintained, which include the engine, 4 brake pads (front left, front right, rear left and rear right, respectively), 4 tyres and 4 springs. They are used as the condition for maintenance in the fixed-interval maintenance scheduling approach, *i.e.* if a component reaches its corresponding average mileage, it is sent for maintenance.

5.2. Experimental results

Table 1 shows the results from the simulator. The first column shows which algorithm has been applied. To optimize the maintenance schedule, four different optimization algorithms have been applied and compared. The basic and preference-based algorithms only take into account three objectives: cost, time and the number of failures. The dynamic basic and dynamic preference-based algorithms involve a fourth objective (the stability of the schedule). This means that the basic and preference-based algorithms handle multi-objective optimization problems, and the dynamic basic and dynamic preference-based algorithms deal with many-objective optimization problems (Purshouse and Fleming 2003), which are more challenging owing to the high computational cost resulting from increased evaluation of the number of points required for PF approximation.

The other columns give the final results of the number of failures (#defects), the total cost (cost), the total maintenance time (time), the number of changed schedules (#ch-sch), the number of unsatisfied trips (#un-trips) and the number of scheduled maintenance activities (#sch-act). Since the maintenance schedule is based on the average mileage and is not updated for fixed-interval maintenance, the number of changed schedules is not applicable in this case. In the table, 'schedule-update: monthly; #evaluations: 100,000" refers to the scenario when the maintenance schedule is updated monthly and the computing budget of the optimization algorithm is 100,000 . All experimental data are the average results from five runs; in each run a different seed for the simulation is used.

After comparing the experimental results, it can be seen that, when there are more workshops, the maintenance time can be reduced because there is less chance that vehicles have to wait for their maintenance. This results in a decrease in the number of unsatisfied trips because the waiting time in workshops is now used to execute trips. Accordingly, the number of maintenance jobs (both the number of scheduled maintenance activities and the number of defects) increases.

When comparing the results of dynamic algorithms with four objectives and their corresponding algorithms without the fourth objective, it can be seen that dynamic algorithms can always reduce the number of changed schedules, but this also means they have to sacrifice the other objectives to some extent. In some industrial scenarios, the stability objective plays a critical role, for example in the case of aircraft maintenance. Some maintenance activities are conducted during the intervals between takeoffs and landings; a change to the maintenance schedule may have an impact on the schedule of the flight and might also disrupt other flights, and rescheduling typically causes significant communication costs.

Next, with more computing budget for the optimization algorithms (*i.e.* the number of objective function evaluations is 500,000), it can be seen that the overall results are improved for three-objective optimization (*i.e.* for basic and preference-based algorithms). The results here refer to the objectives that the algorithms optimize. However, for the dynamic algorithm, results with more computing budget are sometimes mutually non-dominated with the results from using a smaller computing budget. For example, the number of defects can be reduced with the larger computing budget, but the total maintenance cost cannot get improved by more computing budget. This is led by the complexity of many-objective optimization. When determining the schedule to be deployed from the PF, the knee point is chosen. However, in four-dimensional space, a small variation can lead to a big impact on the

Table 1. Optimization results of different maintenance scenarios over five runs.

Twenty vehicles and two workshops						
Algorithm	#defects	cost	time	#ch-sch	#un-trips	#sch-act
Fixed-interval	226	474,965	6,269	N/A	212,450	52
	schedule-update: monthly; #evaluations: 100,000;					
Basic	46	680,666	4,282	4,509	121,000	148
Preference	50	690,871	4,179	4,630	112,150	150
Dynamic basic	73	676,149	5,510	4,023	175,800	154
Dynamic preference	66	688,934	4,732	3,729	137,200	159
	schedule-update: monthly; #evaluations: 500,000;					
Basic	39	675,374	3,936	4,553	101,750	150
Preference	40	677,331	3,903	4,526	101,950	145
Dynamic basic	68	717,046	5,262	3,777	161,300	157
Dynamic preference	42	690,131	4,669	3,240	140,750	150
	schedule-update: weekly; #evaluations: 100,000;					
Basic	32	624,078	4,565	22,884	126,200	166
Preference	35	646,117	4,103	23,016	109,700	168
Dynamic basic	67	633,854	5,758	19,660	185,450	150
Dynamic preference	50	628,228	4,626	18,049	140,200	161
Twenty vehicles and five workshops						
Algorithm	#defects	cost	time	#ch-sch	#un-trips	#sch-act
Fixed-interval	330	747,104	2,996	NA	72,750	92
	schedule-update: monthly; #evaluations: 100,000;					
Basic	68	785,777	1,852	4,877	27,950	192
Preference	67	748,044	1,837	5,012	25,750	184
Dynamic basic	137	849,203	2,942	4,466	62,700	218
Dynamic preference	93	789,592	2,331	4,247	42,000	217
	schedule-update: monthly; #evaluations: 500,000;					
Basic	55	756,278	1,725	4,901	24,850	182
Preference	50	718,176	1,649	4,924	22,550	177
Dynamic basic	125	831,775	2,754	4,442	56,950	223
Dynamic preference	91	797,258	2,130	4,014	34,750	210
	schedule-update: weekly; #evaluations: 100,000;					
Basic	60	695,181	1,995	23,973	35,550	206
Preference	56	690,720	1,951	23,982	31,250	205
Dynamic basic	114	768,697	3,122	21,715	77,950	217
Dynamic preference	91	721,193	2,296	20,397	44,100	227

final result, especially on the accumulated results of multiple optimizations. With monthly schedule updates, the optimization algorithm is executed 22 times during one simulation run of 700 days.

When the schedule is updated more often, *i.e.* weekly, a reduction of the defect number is observed. Apparently, updating the maintenance schedule more frequently can promote its accuracy because the predicted RUL is more accurate. At the same time, an improvement in the total cost can be seen. The reason for the reduction of the total cost also comes from the accuracy of the schedule and the resulting decrease in the penalty cost arising when the vehicle is maintained before it breaks down, *i.e.* the cost of too-early maintenance. When updating the maintenance schedule more often, the maintenance time cannot always be improved because the number of maintenance tasks is not always decreased; the number of maintenance tasks may increase owing to the accuracy of the schedule and the resulting increase in the number driving tasks that have been executed.

When comparing the preference-based algorithm to the basic algorithm, for both three-objective and four-objective optimization, it can be seen that the results of the preference-based algorithm are usually better than those of its corresponding basic algorithm for the scenario of five workshops. However, if there are only two workshops, the waiting of vehicles for their maintenance results in a performance degradation of the preference-based algorithms. The similar working tasks of the vehicles lead to the phenomenon that the scheduled maintenance times for some vehicles are close, and this leads to the situation that the workshops run out of capacity sometimes but become idle at other times. Therefore, a good solution would be to offer more workshops for the fleet; at the same time, these workshops can also work for other tasks besides for the fleet.

Lastly, when comparing with fixed-interval maintenance, there are more defects, maintenance time and unsatisfied trips with fixed-interval maintenance. Since most maintenance tasks are caused by defects, the too-early maintenance cost drops dramatically and this leads to a decrease of the total cost.

5.3. Adjust the execution window

Besides the parameters that change the experimental environment, the variables in the optimization algorithm can also be adjusted to emphasize some aspects of the results. To reduce the number of failures of vehicles, the interval of the execution window (see Section 3) is switched from $[\mu - 2 \times \sigma, \mu + 2 \times \sigma]$ to $[\mu - 3 \times \sigma, \mu + \sigma]$. Through the simulator results after shifting the execution window forward (see the online supplemental data), a dramatic drop in the number of defects is achieved and the descent rate reaches 83.21% on average.

5.4. Increase problem scale

It is worth noting that the problems with 20 vehicles and 13 components for each vehicle are already large scale scheduling optimization problems in terms of the domain of flexible job shop scheduling optimization (FJSS) (Garey, Johnson, and Sethi 1976). The MOVFMSO problem is more complex than FJSS because it needs to assign not only the workshops and maintenance times for the maintenance activities, but also the combination of components for each activity. To investigate how scalable the proposed approach is, the questions asked are whether the algorithms can be applied to even larger fleets and whether consistent results can be achieved when the fleet becomes significantly larger. To this end, the fleet size was increased to 50 vehicles with 15 available workshops, the number of components to be maintained remaining the same. Through the simulator results (see the online supplemental data), it can be observed that the results correspond with the results for 20 vehicles.

6. Summary and outlook

In this work, tailored MOEAs are used to solve a real-world MOVFMSO problem. Compared to most scheduling optimization algorithms in the literature, which only deal with bi-objective optimization, three and four objectives are considered in the proposed algorithms. The maintenance schedule is optimized based on the RUL of components which is predicted based on the actual situation of vehicles to be maintained. Moreover, the preference-based algorithm makes it possible to zoom into the preference region and achieve more fine-grained solutions in this region. Furthermore, when optimization algorithms are required to update maintenance schedules regularly in a dynamic environment, they are extended to dynamic many-objective evolutionary algorithms that take stability as a fourth objective with the aim of maintenance schedule robustness. Another important contribution of this article is the development of the vehicle fleet maintenance optimization simulator, which can be used as a scalable benchmark for optimizing vehicle fleet maintenance schedules in an industrially relevant setting. The simulator and benchmark problems have been inspired by the instances faced by a taxi company with up to 50 vehicles. The proposed MOEAs can be compared and tested easily

in the simulator in a rolling-horizon fashion. Parameters and algorithms can be adjusted to imitate various scenarios. Therefore, although the implementation of the approach is demonstrated by the example of a taxi fleet, the proposed approach can be adapted to different industrial applications, for example the maintenance of trucks, vessels, aircraft, *etc.*

The size of problems in the experiments is up to 50 vehicles and 13 components for each vehicle. Still, one might imagine problems of even larger scale, and finding the fleet size limit that the algorithm can handle would be an interesting subject for future research. For this, the method of accelerating the approach, *e.g.* via surrogate models, might be used. In this article, to maintain clarity of presentation, the dynamically changing element has so-far been restricted, but in future work additional dynamic elements and uncertainties should be considered. For example, uncertainty in the maintenance duration could be modelled, as in Golpîra and Tirkolaee (2019), the presence of cost uncertainty as in Delgoshaei *et al.* (2016), *etc.* Furthermore, techniques to deal with different types of uncertainty, *e.g.* epistemic uncertainty and aleatory uncertainty, could be investigated. For example, Dempster–Shafer theory (uncertainty in probabilistic model parameters), fuzzy modelling (Tirkolaee, Goli, and Weber 2020), or self-learning models that adapt the probability distribution during the execution of the dynamical algorithm, *e.g.* by reinforcement learning. The advantage of fuzzy modelling compared with probabilistic modelling would be that linguistic concepts can be incorporated, which is especially interesting in group decision making (Haque *et al.* 2020) and when decisions are based on expert opinions. However, an in-depth discussion of linguistic versus probabilistic models for maintenance scheduling is beyond the scope of a single article. Certainly, it would be a valuable extension of this work.

Notes

1. CarMaker simulation was developed by IPG Automotive GmbH for testing driving scenarios of passenger cars and light-duty vehicles. It provides models for vehicles, roads, drivers and traffic for all simulation tasks in realistic driving scenarios. See `https://ipg-automotive.com/products-services/simulation-software/carmaker/#driver`
2. https://data.cityofnewyork.us/Transportation/2015-Green-Taxi-Trip-Data/gi8d-wdg5/data

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work is part of the research programme Smart Industry SI2016 with project name CIMPLO and project number 15465, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

ORCID

Markus Olhofer http://orcid.org/0000-0002-3062-3829

References

Camci, Fatih, Kamal Medjaher, Vepa Atamuradov, and Ashyrmuhammet Berdinyazov. 2019. "Integrated Maintenance and Mission Planning Using Remaining Useful Life Information." *Engineering Optimization* 51 (10): 1794–1809.

Das, Indraneel. 1999. "On Characterizing the 'Knee' of the Pareto Curve Based on Normal–Boundary Intersection." *Structural Optimization* 18 (2–3): 107–115.

Delgoshaei, Aidin, Ahad Ali, Mohd Khairol Anuar Ariffin, and Chandima Gomes. 2016. "A Multi-Period Scheduling of Dynamic Cellular Manufacturing Systems in the Presence of Cost Uncertainty." *Computers & Industrial Engineering* 100: 110–132. doi:10.1016/j.cie.2016.08.010.

Elattar, Hatem, Hamdy K. Elminir, and A. M. Riad. 2016. "Prognostics: A Literature Review." *Complex & Intelligent Systems* 2 (2): 125–154.

Garey, Michael, David Johnson, and Ravi Sethi. 1976. "The Complexity of Flowshop and Jobshop Scheduling." *Mathematics of Operations Research* 1 (2): 117–129.

Golpîra, Hêriş, and Erfan Babaee Tirkolaee. 2019. "Stable Maintenance Tasks Scheduling: A Bi-Objective Robust Optimization Model." *Computers & Industrial Engineering* 137: Article ID 106007. doi:10.1016/j.cie.2019.106007.

Haque, Tipu Sultan, Avishek Chakraborty, Sankar Prasad Mondal, and Shariful Alam. 2020. "Approach to Solve Multi-Criteria Group Decision-Making Problems by Exponential Operational Law in Generalised Spherical Fuzzy Environment." *CAAI Transactions on Intelligence Technology* 5 (2): 106–114.

Purshouse, Robin, and Peter Fleming. 2003. "Evolutionary Many-Objective Optimisation: An Exploratory Analysis." In *Proceedings of the Congress on Evolutionary Computation (CEC'03)*, Vol. 3, 2066–2073. Piscataway, NJ: IEEE.

Sobczyk, Kazimierz, and Billie Spencer Jr. 1993. *Random Fatigue: From Data to Theory*. San Diego, CA: Academic Press.

Thiele, Lothar, Kaisa Miettinen, Pekka Korhonen, and Julian Molina. 2009. "A Preference-Based Evolutionary Algorithm for Multi-Objective Optimization." *Evolutionary Computation* 17 (3): 411–436.

Thu Bui, Lam, and Sameer Alam. 2008. *Multi-Objective Optimization in Computational Intelligence: Theory and Practice*. Hershey, PA: Information Science Reference. doi:10.4018/978-1-59904-498-9.

Tinga, Tiedo. 2013. *Principles of Loads and Failure Mechanisms: Applications in Maintenance, Reliability and Design*. London: Springer-Verlag.

Tirkolaee, Erfan Babaee, Alireza Goli, and Gerhard-Wilhelm Weber. 2020. "Fuzzy Mathematical Programming and Self-Adaptive Artificial Fish Swarm Algorithm for Just-In-Time Energy-Aware Flow Shop Scheduling Problem with Outsourcing Option." *IEEE Transactions on Fuzzy Systems* 28 (11): 2772–2783.

Vachtsevanos, George, Frank L. Lewis, Michael Roemer, Andrew Hess, and Biqing Wu. 2006. *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. Hoboken, NJ: Wiley. doi:10.1002/9780470117842.

Van Nguyen, Duc, Marios Kefalas, Kaifeng Yang, Asteris Apostolidis, Markus Olhofer, Steffen Limmer, and Thomas Bäck. 2019. "A Review: Prognostics and Health Management in Automotive and Aerospace." *International Journal of Prognostics and Health Management* 10 (2): 1–35.

Van Nguyen, Duc, Steffen Limmer, Kaifeng Yang, Markus Olhofer, and Thomas Bäck. 2019. "Modeling and Prediction of Remaining Useful Lifetime for Maintenance Scheduling Optimization of a Car Fleet." *International Journal of Performability Engineering* 15 (9): 2318–2328. doi:10.23940/ijpe.19.09.p4.23182328.

Wang, Yali, Michael Emmerich, André Deutz, and Thomas Bäck. 2019. "Diversity-Indicator Based Multi-Objective Evolutionary Algorithm: DI-MOEA." In *International Conference on Evolutionary Multi-Criterion Optimization*, edited by Kalyanmoy Deb, Erik Goodman, Carlos A. Coello Coello, Kathrin Klamroth, Kaisa Miettinen, Sanaz Mostaghim, and Patrick Reed, 346–358. Cham, Switzerland: Springer.

Wang, Yali, Steffen Limmer, Markus Olhofer, Michael T. M. Emmerich, and Thomas Bäck. 2019. "Vehicle Fleet Maintenance Scheduling Optimization by Multi-Objective Evolutionary Algorithms." In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, 442–449. Piscataway, NJ: IEEE. doi:10.1109/CEC.2019.8790142.

Wang, Yali, Steffen Limmer, Markus Olhofer, Michael Emmerich, and Thomas Bäck. 2021. "Automatic Preference Based Multi-Objective Evolutionary Algorithm on Vehicle Fleet Maintenance Scheduling Optimization." arXiv:2101.09556.

Xia, Tangbin, Yifan Dong, Lei Xiao, Shichang Du, Ershun Pan, and Lifeng Xi. 2018. "Recent Advances in Prognostics and Health Management for Advanced Manufacturing Paradigms." *Reliability Engineering & System Safety* 178: 255–268.